H 5

```
EEEEEEEEEEEEEEE  MMM         MMM  UUU         UUU  LLL                 AAAAAAAAA     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  MMM         MMM  UUU         UUU  LLL                 AAAAAAAAA     TTTTTTTTTTTTTTT
EEEEEEEEEEEEEEE  MMM         MMM  UUU         UUU  LLL                 AAAAAAAAA     TTTTTTTTTTTTTTT
EEE              MMMMMM   MMMMMM  UUU         UUU  LLL              AAA         AAA        TTT
EEE              MMMMMM   MMMMMM  UUU         UUU  LLL              AAA         AAA        TTT
EEE              MMMMMM   MMMMMM  UUU         UUU  LLL              AAA         AAA        TTT
EEE              MMM   MMM   MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEE              MMM   MMM   MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEE              MMM   MMM   MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEEEEEEEEEEE     MMM         MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEEEEEEEEEEE     MMM         MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEEEEEEEEEEE     MMM         MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEE              MMM         MMM  UUU         UUU  LLL              AAAAAAAAAAAAAAA        TTT
EEE              MMM         MMM  UUU         UUU  LLL              AAAAAAAAAAAAAAA        TTT
EEE              MMM         MMM  UUU         UUU  LLL              AAAAAAAAAAAAAAA        TTT
EEE              MMM         MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEE              MMM         MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEE              MMM         MMM  UUU         UUU  LLL              AAA         AAA        TTT
EEEEEEEEEEEEEEE  MMM         MMM  UUUUUUUUUUUUUUU  LLLLLLLLLLLLLLL  AAA         AAA        TTT
EEEEEEEEEEEEEEE  MMM         MMM  UUUUUUUUUUUUUUU  LLLLLLLLLLLLLLL  AAA         AAA        TTT
EEEEEEEEEEEEEEE  MMM         MMM  UUUUUUUUUUUUUUU  LLLLLLLLLLLLLLL  AAA         AAA        TTT
```

```
_$2



Sym
---
CMP
DEC
DEC
DEC
DEC

DEC
EXE
EXE
EXE
MMG
MMG
MOV
PR$
SYS
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
VAX
```

```
**FILE**ID**VAXLOAD                                          C  1

VV      VV    AAAAAA    XX      XX  LL            OOOOOO      AAAAAA    DDDDDDDD
VV      VV    AAAAAA    XX      XX  LL            OOOOOO      AAAAAA    DDDDDDDD
VV      VV  AA      AA  XX      XX  LL          OO      OO  AA      AA  DD      DD
VV      VV  AA      AA  XX      XX  LL          OO      OO  AA      AA  DD      DD
VV      VV  AA      AA    XX  XX    LL          OO      OO  AA      AA  DD      DD
VV      VV  AA      AA    XX  XX    LL          OO      OO  AA      AA  DD      DD
VV      VV  AA      AA      XX      LL          OO      OO  AA      AA  DD      DD
VV      VV  AA      AA      XX      LL          OO      OO  AA      AA  DD      DD
VV      VV  AAAAAAAAAA    XX  XX    LL          OO      OO  AAAAAAAAAA  DD      DD
VV      VV  AAAAAAAAAA    XX  XX    LL          OO      OO  AAAAAAAAAA  DD      DD
  VV  VV    AA      AA  XX      XX  LL          OO      OO  AA      AA  DD      DD      ....
  VV  VV    AA      AA  XX      XX  LL          OO      OO  AA      AA  DD      DD      ....
    VV      AA      AA  XX      XX  LLLLLLLLLL    OOOOOO    AA      AA  DDDDDDDD        ....
    VV      AA      AA  XX      XX  LLLLLLLLLL    OOOOOO    AA      AA  DDDDDDDD        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
0000          .NLIST  CND
0000          .TITLE  VAX$LOAD - HEADER FOR LOADABLE CHAR/DECIMAL EMULATION
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000    ;++
0000    ;
0000    ; Facility:
0000    ;
0000    ;       Instruction Emulator
0000    ;
0000    ; Abstract:
0000    ;
0000    ;       This module defines the data structures required for a piece
0000    ;       of loadable code.  This includes the pool header and the code
0000    ;       needed to hook into the rest of the system.  For the instruction
0000    ;       emulation code, the hooks are vectors in the SCB.
0000    ;
0000    ; Environment: MODE=Kernel
0000    ;
0000    ; Author:  Kathleen D. Morse, Creation date:  04-May-1983
0000    ;
0000    ; Modified by:
0000    ;
0000    ;       V03-004 LJK0028         Lawrence J. Kenah       10-Apr-1984
0000    ;               Store base address of emulator image in cell in SYS.EXE
0000    ;               set aside for that purpose.
0000    ;
0000    ;       V03-003 LJK0027         Lawrence J. Kenah       21-Mar-1984
0000    ;               Store address of access violation handler into EXE$GL_VAXEXCVEC
0000    ;               when loading decimal/string emulator.
0000    ;
0000    ;       V03-002 LJK0017         Lawrence J. Kenah       17-Jan-1984
0000    ;               Make table entries for SCB entries position independent.
```

```
                  0000  ;               Change PSECT attributes.
                  0000  ;
                  0000  ;       V03-001 WMC0001         Wayne Cardoza           23-Jun-1983
                  0000  ;               Fix SLVTAB.
                  0000  ;
                  0000  ;--
                  0000
                  0000  ;
                  0000  ; INCLUDE FILES:
                  0000  ;
                  0000
                  0000          $PRTDEF                                 ; Define protection codes
                  0000          $PTEDEF                                 ; Define page table entry fields
                  0000          $VADEF                                  ; Define virtual address fields
                  0000
                  0000  ; This must be the first program section in the image file.
                  0000
              00000000          .PSECT  $$$$$$BEGIN     PAGE,PIC,USR,CON,REL,GBL,SHR,NOWRT
                  0000
                  0000          .ENABLE         LOCAL_BLOCK
                  0000
                  0000  10$:
                  0000
                  0000  VAX$BEGIN::                                     ; Beginning of string/decimal emulator
                  0000          SLVTAB  END=VAX$END , -                 ; Size of string/decimal ins emulator
                  0000                  INITRTN=VAX$INIT , -            ; Address of initialization routine
                  0000                  SUBTYP=DYN$C_NON_PAGED, -       ; Sub-type for data structure
                  0000                  PROT_W=PRT$C_URKW , -           ; Protection on loadable code pages
                  0000                  FACILITY=VAXEMUL.EXE            ; Name of image loaded
                  0024
                  0024
                  0024  SCB_UVAX:                                               ; Hook for SCB uVAX except
       000000C8   0024          .LONG   ^XC8                                    ;  Offset into SCB
       FFFFFE01'  0028          .LONG   VAX$EMULATE - VAX$BEGIN_UR              ;  Offset to emulator entry pt
                  002C  SCB_UVAX_FPD:                                           ; Hook for SCB uVAX FPD except
       000000CC   002C          .LONG   ^XCC                                    ;  Offset into SCB
       FFFFFE01'  0030          .LONG   VAX$EMULATE_FPD - VAX$BEGIN_UR          ;  Offset to emulator entry pt
       00000000   0034          .LONG   0                                       ; Empty hook ends table
                  0038
       00000038   0038  ...SIZE... = .-10$
                  0038
                  0038  ; Insure at least one page before real code begins
                  0038
                  0038  SPACE_FILLER1:                                          ; This prevents UR access to
       000001FF   0038          .BLKB   <511 - ...SIZE...>                      ;  the pool fragments on either
                  01FF                                                          ;  side of the emulation code.
                  01FF          .DISABLE        LOCAL_BLOCK
                  01FF
                  01FF  VAX$BEGIN_UR::                                          ; Starting VA to protect UR
```

```
                                     01FF          .SBTTL  VAX$INIT - Initialization routine to hook into SCB
                                     01FF
                                     01FF  ;++
                                     01FF  ; Functional Description:
                                     01FF  ;
                                     01FF  ;       VAX$INIT is linked together with all of the code required for
                                     01FF  ;       the instruction emulator.  The necessary amount of non-paged pool
                                     01FF  ;       is allocated and rounded up to page boundary.  Code is then
                                     01FF  ;       moved into this block of pool.  All of this code must be PIC.
                                     01FF  ;       This code is then re-protected so that it can be executed from
                                     01FF  ;       user mode.  A page is allocated on either side of the emulator
                                     01FF  ;       to serve as buffers, because the code is not loaded on a page
                                     01FF  ;       boundary and pool cannot be protected UR for security reasons.
                                     01FF  ;
                                     01FF  ;       The vectors for instruction emulation and instruction emulation
                                     01FF  ;       first-part-done are then connected to the emulation code.
                                     01FF  ;
                                     01FF  ; Calling Sequence:
                                     01FF  ;
                                     01FF  ;       JSB     VAX$INIT
                                     01FF  ;
                                     01FF  ; Input Parameters:
                                     01FF  ;
                                     01FF  ;       None
                                     01FF  ;
                                     01FF  ;--
                                     01FF
                                     01FF  ; This PSECT holds the init routines.
                                     01FF
                                 00000000          .PSECT  ____INITHK      BYTE,PIC,USR,CON,REL,GBL,SHR,NOWRT
                                     0000
                                     0000          .ENABLE         LOCAL_BLOCK
                                     0000
                                     0000  10$:
                                     0000  VAX$INIT::                                      ; Hook in emulation code
                                     0000  VAX$END_UR::                                    ; Also ending VA to protect UR
                  7E     50     7D   0000          MOVQ    R0,-(SP)                        ; Save registers
                  7E     52     7D   0003          MOVQ    R2,-(SP)                        ; Save registers
                                     0006
                                     0006  ;
                                     0006  ; Now reset the protection on the non-paged pool to be
                                     0006  ; user-read, so that the emulation code can be accessed from
                                     0006  ; all modes.  Make it kernel-write so that breakpoints can be
                                     0006  ; set in the emulation code with XDELTA.
                                     0006  ;
          51     01FF'CF            9E   0006          MOVAB   W^VAX$BEGIN_UR,R1               ; Get starting VA to protect URKW
   51     51     15     09          EF   000B          EXTZV   #VA$V_VPN,#VA$S_VPN,R1,R1 ; Make address into VPN
          51     51     02          78   0010          ASHL    #2,R1,R1                        ; Make into byte index into SPT
          52     FFE8 CF            9E   0014          MOVAB   W^VAX$END_UR,R2                 ; Get ending address to protect URKW
   52     52     15     09          EF   0019          EXTZV   #VA$V_VPN,#VA$S_VPN,R2,R2  ; Make address into VPN
          52     52     02          78   001E          ASHL    #2,R2,R2                        ; Make into byte index into SPT
                 50     0E          9A   0022          MOVZBL  #PRT$C_URKW,R0                  ; New protection for emulation code
   53     00000000'GF               9E   0025          MOVAB   G^MMG$GL_SPTBASE,R3             ; Get address of system page table
                                     002C  20$:
00 B341   04     1B     50          F0   002C          INSV    R0,#PTE$V_PROT,#PTE$S_PROT,@(R3)[R1] ; Set new
   FFF3 51       04     52          F1   0033          ACBL    R2,#4,R1,20$                    ; protection for each page
                                     0039          INVALID                                 ; Invalidate the translation buffer
```

```
                             003C   221
                             003C   222 ;
                             003C   223 ; Now connect the emulation code to the system control block.
                             003C   224 ;
  53   00000000'GF    D0     003C   225         MOVL    G^EXE$GL_SCB,R3              ; Base address of SCB
00C8 C3    0000'CF    9E     0043   231         MOVAB   W^VAX$EMULATE,^XC8(R3)       ; Set SCB to point to emulator code
00CC C3    0000'CF    9E     004A   232         MOVAB   W^VAX$EMULATE_FPD,^XCC(R3)   ; Set SCB to point to emulator code
           0000'CF    9E     0051   233         MOVAB   W^VAX$MODIFY_EXCEPTION,-     ; Store address of access violation
       00000000'GF           0055   234                 G^EXE$GL_VAXEXCVEC           ;  handler
           0000'CF    9E     005A   235         MOVAB   W^VAX$BEGIN,-
       00000000'GF           005E   236                 G^MMG$GL_VAXEMUL_BASE        ; Store base address of image
             52   8E  7D     0063   238         MOVQ    (SP)+,R2                   ; Restore registers
             50   8E  7D     0066   239         MOVQ    (SP)+,R0                   ; Restore registers
                      05     0069   240         RSB                                ; and return
                             006A   241
                   0000006A  006A   242 ...INIT_SIZE... = .-10$
                             006A   243
                             006A   244         .DISABLE        LOCAL_BLOCK
                             006A   245
                             006A   246 ; This must be the last program section in the image
                             006A   247
                         00000000   248         .PSECT  ______END       BYTE,PIC,USR,CON,REL,GBL,SHR,NOWRT
                             0000   249
                             0000   250 ; Insure at least one page at the end of the image, too
                             0000   251
                             0000   252 SPACE_FILLER2:                                  ; This prevents UR access to
                   00000195  0000   253         .BLKB   <511 - ...INIT_SIZE...>         ;  the pool fragments on either
                             0195   254                                                 ;  side of the emulation code.
                             0195   258 VAX$END::                          ;
                             0195   260
                             0195   261         .END
```

```
...INIT_SIZE...                 = 0000006A
...SIZE...                      = 00000038
DYN$C_LOADCODE                  = 00000062
DYN$C_NON_PAGED                 = 00000001
EXE$GL_SCB                        ********    X    03
EXE$GL_VAXEXCVEC                  ********    X    03
MMG$GL_SPTBASE                    ********    X    03
MMG$GL_VAXEMUL_BASE               ********    X    03
PR$_TBIA                          ********    X    03
PRT$C_ER                        = 00000007
PRT$C_URKW                      = 0000000E
PTE$S_PROT                      = 00000004
PTE$V_PROT                      = 0000001B
SCB_UVAX                          00000024 R       02
SCB_UVAX_FPD                      0000002C R       02
SPACE_FILLER1                     00000038 R       02
SPACE_FILLER2                     00000000 R       04
VA$S_VPN                        = 00000015
VA$V_VPN                        = 00000009
VAX$BEGIN                         00000000 RG      02
VAX$BEGIN_UR                      000001FF RG      02
VAX$EMULATE                       ********    X    02
VAX$EMULATE_FPD                   ********    X    02
VAX$END                           00000195 RG      04
VAX$END_UR                        00000000 RG      03
VAX$INIT                          00000000 RG      03
VAX$MODIFY_EXCEPTION              ********    X    03

                                         +----------------+
                                         ! Psect synopsis !
                                         +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$$$$BEGIN                     000001FF  (  511.)  02 (  2.)    PIC   USR   CON   REL   GBL   SHR   EXE   RD   NOWRT NOVEC PAGE
____INITHK                      0000006A  (  106.)  03 (  3.)    PIC   USR   CON   REL   GBL   SHR   EXE   RD   NOWRT NOVEC BYTE
______END                       00000195  (  405.)  04 (  4.)    PIC   USR   CON   REL   GBL   SHR   EXE   RD   NOWRT NOVEC BYTE

                                     +------------------------+
                                     ! Performance indicators !
                                     +------------------------+

Phase                     Page faults    CPU Time       Elapsed Time
-----                     -----------    --------       ------------
Initialization                     18    00:00:00.05    00:00:01.56
Command processing                 86    00:00:00.48    00:00:05.97
Pass 1                            127    00:00:03.01    00:00:11.34
Symbol table sort                   0    00:00:00.27    00:00:00.46
Pass 2                             41    00:00:00.69    00:00:04.21
Symbol table output                 4    00:00:00.04    00:00:00.04
Psect synopsis output               2    00:00:00.02    00:00:00.02
Cross-reference output              0    00:00:00.00    00:00:00.00
Assembler run totals              278    00:00:04.56    00:00:23.61

The working set limit was 900 pages.
```

```
16957 bytes (34 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 250 non-local and 5 local symbols.
261 source lines were read in Pass 1, producing 17 object records in Pass 2.
12 pages of virtual memory were used to define 11 macros.

                                        +----------------------------+
                                        ! Macro library statistics !
                                        +----------------------------+

Macro library name                      Macros defined
------------------                      --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                       5
_$255$DUA28:[SYSLIB]STARLET.MLB;2                    4
TOTALS (all libraries)                               9

355 GETS were required to define 9 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:VAXLOAD/OBJ=OBJ$:VAXLOAD MSRC$:LOADHDR/UPDATE=(ENH$:LOADHDR)+EXECML$/LIB
```

0145 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY